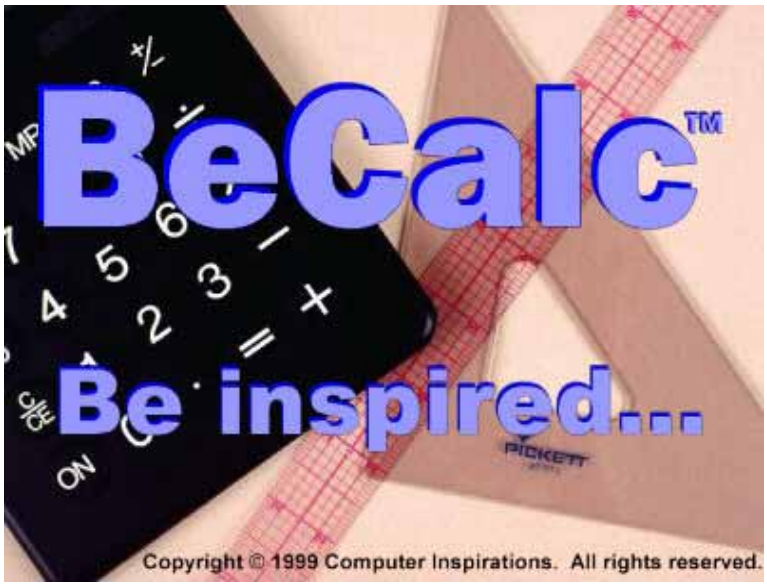


User Manual

Installation Instructions
And Guided Tutorials

BeCalc™ is a trade mark of Computer Inspirations.
All other brand and product names are trade marks of their respective owners.
Copyright © 1999 by Computer Inspirations
All rights reserved.

Table of Contents

Introduction 1
- What is BeCalc? 1
- Uses For BeCalc 2
- Conventions 2
- Drag and Drop 2
- Single-Level Undo 3
- Window, Font, and Color Settings 3
- Other Automatic Settings 3

Installation 5
- Minimum Installation Requirements 5
- Automated Installation 5
- Installation Hierarchy 6

Tutorials 7
- Overview 7
- Numerical Range and Accuracy 8
- Enter and Evaluate an Equation 8
- Complex Numbers 10
- Metric Conversions 11
- Mortgage Calculation 13
- Printing a Mortgage List 13
- Exporting a Mortgage List 13
- Plotting a Graph 14
- Exporting a Graph 15
- Statistics Population Characteristics 16
- Programmer Functions 17
- Radix Number Accuracy 18
- Customizations 19
- Digit Separators 19
- Numerical Format 20
- Equation and Result Settings 21
- Variable Settings 21

Reference 22

List of Figures and Tables

- Figure 1—Overview of the many *BeCalc* functions and operators 1
- Figure 2—*BeCalc* installation components 6
- Figure 3—Organization of *BeCalc*'s graphical interface controls 8
- Figure 4—Evaluate sample and hold time equation: $\tau_2 \cdot e^{\frac{\tau_b}{\tau_1}} \cdot \left(1 - e^{\frac{\tau_b - t}{\tau_2}} \right)$ 9
- Figure 5—Converting miles per gallon to liters per 100 Km11
- Figure 6—Mortgage with $20000 amortized for 48 months at 4% per year 13
- Figure 7— Plot of the equation $e^{\frac{-\pi x}{10}} \cdot \sin \pi x$ 14
- Figure 8—Statistical Functions 16
- Figure 9—Digit Separators 19
- Figure 10—Numerical Format 20
- Figure 11—Equation and Result Settings 21
- Figure 12—Variable Settings 21

- Table 1—BeCalc available operations and operators 4
- Table 2—Control Key Special Characters 9
- Table 3—Karnaugh map of Y = (A ⊕ B) + B·C + A·D + (B ⊕ D) 18

Chapter

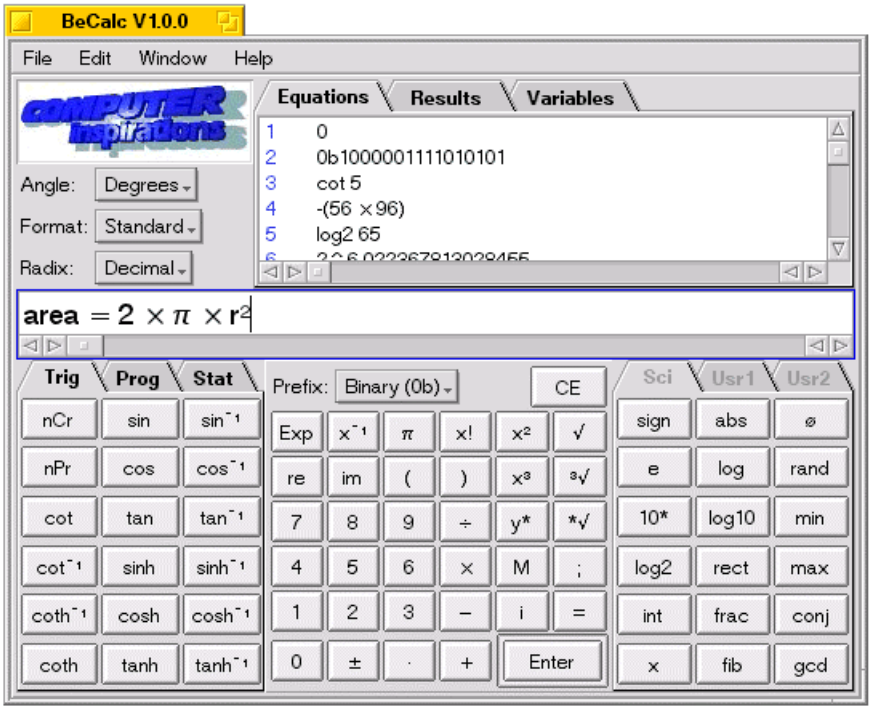
1

BeCalc User Manual

Installation Instructions And Guided Tutorials

Introduction

With hundreds of different calculators available today—why Yet Another Calculator? Most calculators mimic the same inane interface that forces you to enter numbers, followed by a click on an operator and entry of the final number. It is easy to become lost halfway through the entry of a particularly long equation and then have to start all over again. After displaying an answer, if you need to repeat the calculation, guess what—you re-enter the whole equation again...and again. Now there's a better way. *BeCalc* provides you with the tools to surmount this and many other failings of traditional calculators.



• Figure 1—Overview of the many *BeCalc* functions and operators

What is BeCalc?

BeCalc is an expression-evaluating calculator with a graphical interface (**Error! Reference source not found.**). Equations are entered using either mouse clicks on the buttons or are typed in directly from either the keyboard or numeric keypad. Non-volatile history lists

store equations and results which are accessible with a click of the mouse. Arbitrary, named variables, used in calculations, reside in a third scrollable list. The output numbers use standard floating point notation, fixed point, scientific or engineering notation. Furthermore, calculations support imaginary numbers for most functions and are accurate to more than 200 digits. Logical functions work with binary, octal, decimal, hexadecimal and other based numbers with accuracies of more than 670 bits. Table 1 displays the available operators and operations.

BeCalc also plots two-dimensional graphs of entered equations, creates mortgage tables, analyzes statistical trends, and converts between metric, English, and US units. For a complete listing of all *BeCalc's* capabilities refer to ***Chapter 3***, page 7.

Uses For BeCalc

Why use *BeCalc* instead of a spreadsheet?

- Convenient hexadecimal or binary calculations;
- More accuracy for computing with 64- and 128-bit numbers;
- Quick comparison of different mortgage interest rates;
- Perform simple metric and other conversions;
- Function-based plots without requiring rows of data and "wizards";
- Shows equations instead of just the results of the calculations;
- Keeps a history of calculations—useful for accounting and audit purposes;
- Gives extremely accurate results (>200 digits) for validating floating point routines;
- Improved readability with named variables in equations;
- Automated statistical trends analysis;
- Saves money over the cost of a spreadsheet.

Conventions

In this manual we use the user interface conventions adopted by the BeOS User Guide. Refer to your guide for details.

Drag and Drop

BeCalc supports the standard BeOS drag and drop operations in many places:

1) Color drops to windows and lists from applications such as the freely-available ColorSelector, etc. change the background color where the color selection was dropped.
2) Font drops to windows and lists from applications such as the freely-available FontSelector, etc. change the font of the window or list where the font selection was dropped. The window can be resized to properly display larger fonts.
3) Any of the equations, results, or variables may be dragged from the list and dropped to the entry control at any position. Alternatively, these items may be dragged to other applications which support text-based drops.
4) The entry control text can be selected and dragged to other applications or text from another application can be dropped into the entry control at any point.
5) Plots can be dragged as bitmaps into other applications such as bitmap editors to customize the plots and then print them. Alternatively, drag plots to the desktop or a

directory to save them as a file for future reference. Note: None of the applications tested have supported the dragging of graphics using Be's recommended protocol.

Single-Level Undo

The entry control supports the standard cut, copy, and paste operations. Most operations on the entry control are undoable with a single-level. Subsequent undos will simply redo the previous action.

Window, Font, and Color Settings

All changes to the window colors and fonts are automatically saved when *BeCalc* exits. Settings are restored the next time *BeCalc* starts.

Other Automatic Settings

BeCalc also remembers the current selection of all tabs, pop-up menus, and controls in all the windows. If any windows are open when *BeCalc* exits, they will be reopened the next time *BeCalc* starts.

Operator	Description
+	Addition
-	Subtraction
***, ×**	Multiplication
/, ÷	Division
^{2}, ^{3}	Squared, Cubed
$^{-1}$	Reciprocal
()	Brackets
^, **	Power
&, and	Logical and
|, or	Logical inclusive or
xor	Logical exclusive or
~, not	Logical complement
mod	Modulo
div	Integer division
√, sqrt	Square root
^{3}√, cbrt	Cube root
***√, root**	Any root
rand	Random number between 0 and 1
fib	The nth Fibonacci number
gcd (<x>; <y>)	Greatest common denominator of x and y
e	Natural logarithm base
i	Imaginary number
r, ø	Complex magnitude, angle
re, im	Real, imaginary parts of a number
rect	Polar to rectangle conversion
conj	Complex conjugate
conv <n>, convi <n>	Convert n units to/from destination units
nCr, nPr	Factorial combinations & permutations
int, frac	Integer and fractional parts of a number
abs, sign	Absolute value, sign of a number (-1 or 1)
min(<x>;<y>), max(<x>;<y>)	Minimum and maximum of x and y
ln, log, log2, log10	Natural (base e), base 2, and base 10 logarithms
sin, asin, sin^{-1}	Sine, Arcsine
cos, acos, cos^{-1}	Cosine, Arccosine
tan, atan, tan^{-1}	Tangent, Arctangent
cot, acot, cot^{-1}	Cotangent, Arccotangent
sinh, asinh, sinh^{-1}	Hyperbolic Sine, Arcsine
cosh, acosh, cosh^{-1}	Hyperbolic Cosine, Arccosine
tanh, atanh, tanh^{-1}	Hyperbolic Tangent, Arctangent
coth, acoth, coth^{-1}	Hyperbolic Cotangent, Arccotangent
»x, »x;y	Add to statistical x and/or y population
CS	Clear statistic state
Avg(x), Avg(y)	Display average of x/y population
Σx, Σy, Σxy, Σx^2, Σy^2	Display sum of all x, y, x·y, x^2, y^2 values
Sx, Sy	Display sigma value of x, y populations
Dev(x), Dev(y)	Display the standard deviation in x, y
n	Display the total number of x/y values
cbit, sbit, tbit	clear, set, toggle bit
shl, shr	Shift left, right
asr	Arithmetic Shift Right
rol, ror	Rotate left, right
<var-name>	Variable contents
<var-name> = <exp>	Initialize variable with expression
π, pi	Constant Pi

• Table 1—BeCalc available operations and operators

Chapter

2

BeCalc User Manual

Installation Instructions And Guided Tutorials

Installation

BeCalc installation requires only a few mouse clicks and should generally proceed without any problems. If, however, you encounter difficulties, please read over the following sections for some hints on what might have gone wrong. Generally, if a problem occurs, start over with a fresh install. The uninstallation procedure helps you remove *BeCalc* in such an event. If you have difficulties, also check that the installation hierarchy matches the installed components in your system. If nothing seems to help, contact our support staff at support@comp-inspirations.com and please remember to send a detailed description of what went wrong—including any messages, a directory listing of your /boot/home/config/lib directory, and a description of your system including CPU type, amount of memory, available hard disk space, and which version of BeOS® you are using.

Minimum Installation Requirements

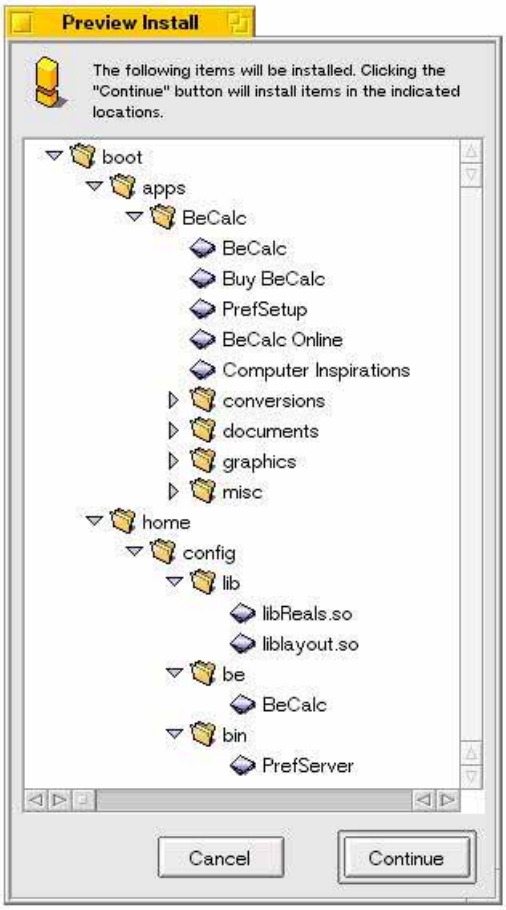
BeCalc can be installed on any system which can run BeOS release 4.5.2. Performance will vary with the kind and number of processors and the amount of memory available. Tests have been performed on a minimal dual processor PPC-based 166MHz BeBox and on a dual processor Pentium II-based 300MHz machine. Due to the current button display implementation, screen refreshes during resizing and window dragging on slower machines may be a big sluggish. The speed of calculations will also vary with the speed of your computer.

Automated Installation

Install *BeCalc* by double-clicking on the package icon. The freely-available SoftWareValet™ from http://www.bedepot.com/Products/Be/SoftwareValet.asp must be installed on your system first. You may install *BeCalc* anywhere although the standard location for BeOS applications is the *apps* directory. As a last installation step, you will be asked whether you would like a link to *BeCalc* placed on the desktop. Doing so gives you instant access to *BeCalc* and is recommended.

Installation Hierarchy

The installation hierarchy is shown in Figure 2 below.



• Figure 2—*BeCalc* installation components

Registration

Register *BeCalc* by double-clicking on the ***Buy BeCalc*** icon. The registration process guides you through a series of forms which capture all the required information regarding your payment method, name, and address and give you a choice of methods to submit your information. Refer to the BeOS Kagi page for more details. Once your payment has been received, you will receive a personalized key file via email which unlocks *your BeCalc* copy.

You may also register online at a secure web site. Refer to our web site for more details. Online transactions require a credit card.

Chapter

3

BeCalc User Manual

Installation Instructions And Guided Tutorials

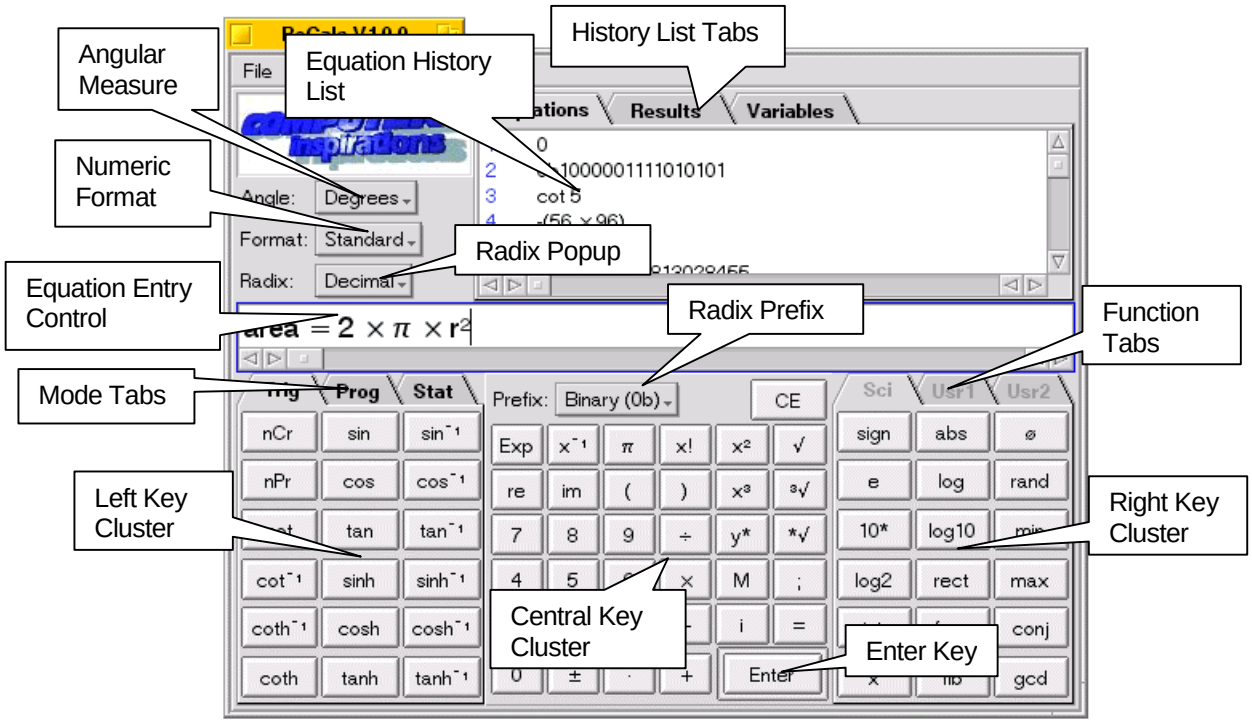
Tutorials

What can I do with this shiny new calculator? Pretty much anything. The following tutorials guide you through but a few of the many uses for *BeCalc*. You will learn how to:

- enter, evaluate, and work with equations and named variables,
- convert to/from metric units,
- prepare, export, and print a mortgage table,
- plot a two-dimensional, labelled graph complete with zooming features,
- calculate statistical characteristics of a two-variable system,
- use binary, octal and hexadecimal arithmetic and conversions,
- customize the number formats and calculator defaults.

Overview

The main interface (see Figure 3) with *BeCalc* is the *equation entry control* which can be used to enter equations either via mouse clicks on the displayed buttons or by entering equations directly into the equation entry control via the keyboard. This control automatically remains focused as long as the main window is active. Each time the *Enter* key is clicked, the active equation is evaluated, stored in the *equation history* list, and the result is displayed. The results are also stored in a *result history* list. A third list, called the *variable* list, gives a snapshot of the state of all user-defined variables. The different lists are displayed by clicking on the *list selection tabs* above the list display. The *left key cluster* also has tabs located above it. Clicking on the *mode select tabs* selects either trigonometric, programmer, or statistical functions. The *right key cluster* has ghosted out tabs located above it. These *function select tabs* are reserved for future expansion where user-defined functions will be supported by *BeCalc*. The *central key cluster* defines the most commonly-used operators and functions.



• Figure 3—Organization of *BeCalc*'s graphical interface controls

Numerical Range and Accuracy

All *BeCalc* operators and functions internally calculate results accurate to more than 200 digits. The actual displayed precision is adjustable using the customization dialogue on page 19. The default display precision is about 16 digits. Numbers have a dynamic range of at least $10^{\pm 14,000,000}$.

Enter and Evaluate an Equation

For example, assume we need to evaluate the equation

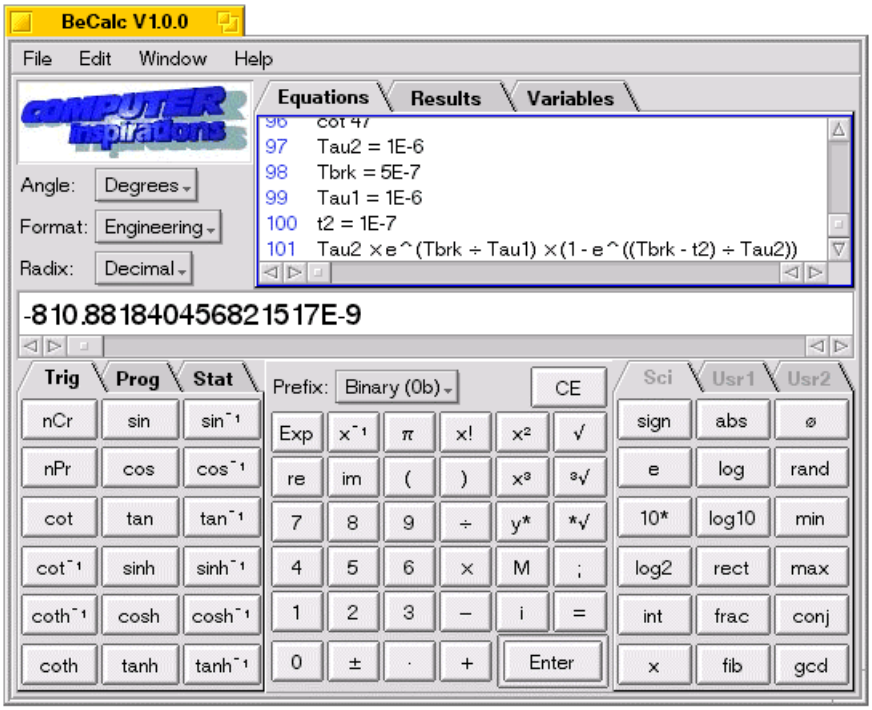
$$\tau_2 \cdot e^{\frac{\tau_b}{\tau_1}} \cdot \left(1 - e^{\frac{\tau_b - t}{\tau_2}}\right)$$

which is a time equation for a multiplexer sample and hold system (see Figure 4). First, we need to define the initial values for the variables. Enter *Tau2 = 1E-6* to define the value of *Tau2* as being 1×10^{-6} where the *=* means the value is stored into the preceding variable. Similarly define *Tbrk* = 5×10^{-7}, *Tau1* = 1×10^{-6} and *t2* = 1×10^{-7}. Then enter the equation followed by an *Enter* or a click on the Enter key.

One of *BeCalc*'s powerful features is the ability to enter variable values only once as the *Tau2* and *Tbrk* values were used in several places. Another feature is the ability to recall a previous equation by dragging it from the history list to the entry control. In this example, it is trivial to change one of the variables: clear the current result with the CE key, drag the equation and click on the Return key to recalculate the result.

To simplify the construction of your initial equation, after you have defined the variables, you can also drag them from the variable list and drop them directly into the entry control to build up the equation without having to retype the variables.

Equations, results, variables, and the contents of the entry control are also draggable into other applications which support standard text drag and drop. In order to display the full range of operators available in *BeCalc* these applications will also need to support the UTF-8 character set which is used for some *BeCalc* operators. There are ASCII equivalents defined for each operator which uses UTF-8 character codes to simplify keyboard equation entry. Table 1 shows these equivalent functions and operators. Table 2 shows the special operators available from the keyboard via control key combinations.



• Figure 4—Evaluate sample and hold time equation: $\tau_2 \cdot e^{\frac{\tau_b}{\tau_1}} \cdot \left(1 - e^{\frac{\tau_b - t}{\tau_2}}\right)$

Desired Character(s)	Key Sequence
÷	<ctrl> /
√	<ctrl> v
π	<ctrl> p
ø	<ctrl> o

• Table 2—Control Key Special Characters

Complex Numbers

Complex numbers are important quantities in engineering and mathematics where they are used in connection with electric circuits, mechanical vibrating systems, fluid dynamics and electrostatics.

BeCalc supports complex number arithmetic in all the standard operators such as addition, subtraction, multiplication, and division. Roots, powers, and both circular and hyperbolic trigonometric functions are also defined for complex numbers. The complex notation used in *BeCalc* is *a + bi* where a, b can be any real numbers. Thus to multiply two imaginary numbers we would enter: (1 + 2i) × (23 – 15i) = 53 + 31i. There is an implied multiplication in the 2i and 15i terms. This is a convenient shorthand which can only be used with this imaginary notation (i.e., it does not apply to variables). It would also be correct, although more tedious, to write (1 + 2 × i) × (23 – 15 × i).

Several convenience functions convert between imaginary and real numbers. The real part of an imaginary number is extracted using the re() function. Similarly, the im() function extracts the imaginary part of a complex number. To convert between imaginary numbers in rectangular format and polar format, the abs() function returns the magnitude of an imaginary number and the ø() function calculates the polar angle of an imaginary number. The rect operator converts a complex number in polar notation to rectangular notation.

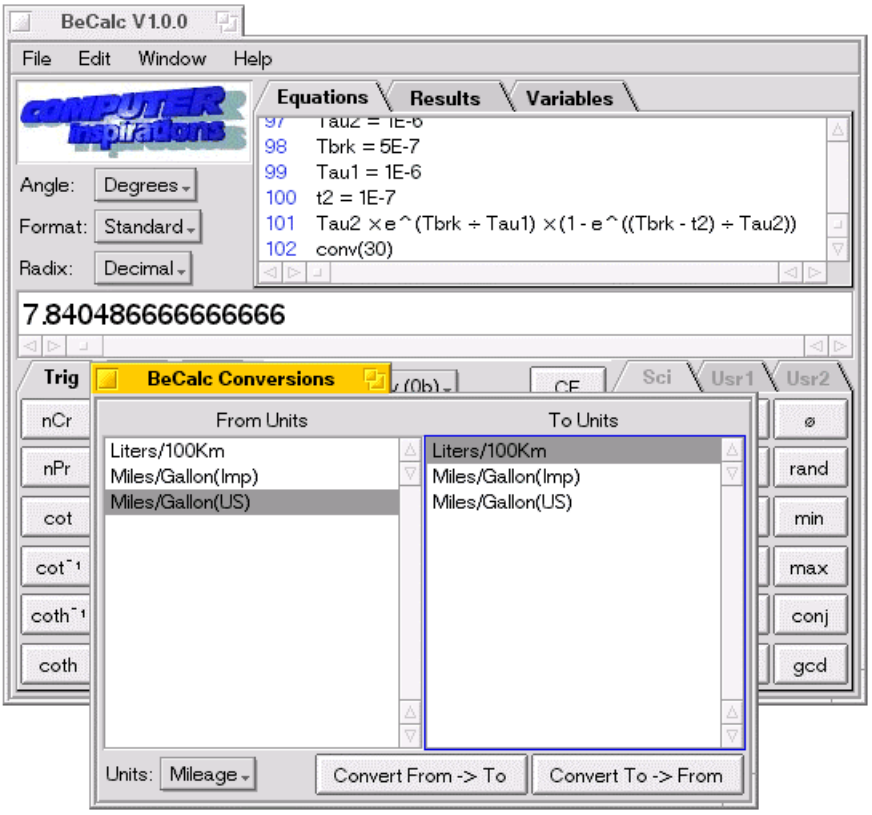
Many of the programmer and statistical functions do not operate on imaginary numbers. They naively take just the real component of the imaginary number for use in their calculations. Other scientific functions such as min, max, fib, int, frac, rand, and sign do not support imaginary numbers and plots will also only graph a range of real values.

Metric Conversions

The USA is the only country in the world which has not adopted the metric system. *BeCalc* provides a convenient method of translating between imperial and metric units to help when dealing with international companies which use metric units. Figure 5 shows the conversion window that opens when the *Unit Conversion* button is clicked. For example, to convert from miles/gallon to 100km/litre, use the popup menu (invoked by holding down the left mouse button) of the *Types* cycle control to select the "mileage" entry. Highlight "miles/gallon" in the *From Units* list and enter "30" in the equation entry control. Click on the *Convert-to-from* button to display the equivalent metric units.

To quickly calculate the inverse operation, just click on the *Convert-from-to* button to convert the displayed number from the highlighted *To Units* list item to the highlighted *From Units* list item.

Whenever possible converted results are exactly accurate and the table conversion factor is marked with an '*'. Other conversions will usually be accurate to at least five digits. When converting between two units, the accuracy will be the lesser of the two.



• Figure 5—Converting miles per gallon to liters per 100 Km

User-Defined Conversions

BeCalc uses a flexible database system based on BeOS attributes to define conversion quantities. Users are able to add additional conversion units and new conversion types and BeCalc automatically detects these new units and types and adds them to the con-

version popup menu and the unit lists. Figure 6 shows the contents of the conversion directory. To add a new conversion unit to an existing conversion type:

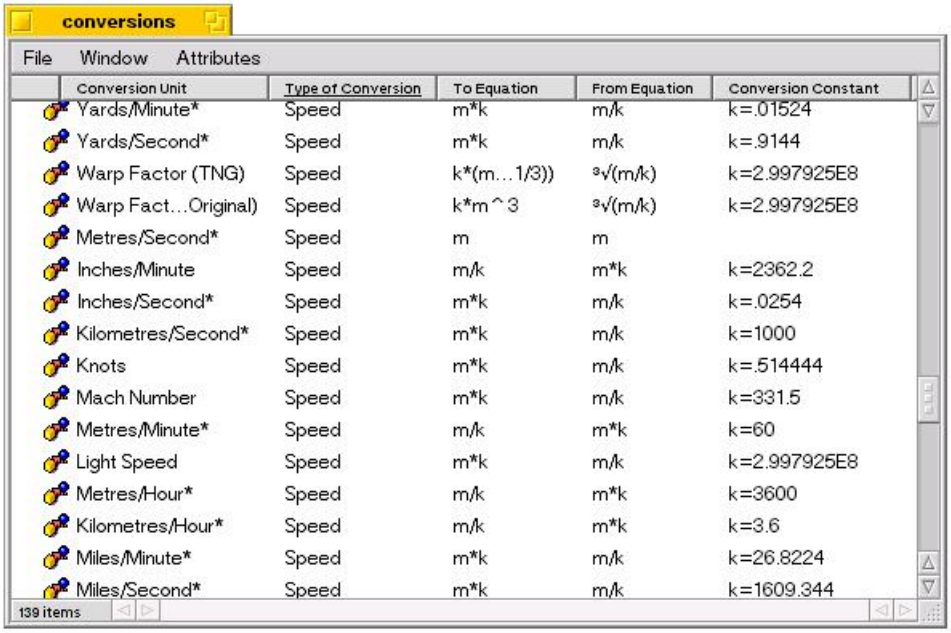
1) Click on an existing conversion unit which is similar to the one being added.
2) From the File menu, select the duplicate item to create a duplicate conversion unit.
3) To edit any of the fields, just click on the field to highlight the text and click again to begin editing the field.

To add a new conversion type (e.g., *Power*):

1) As before, duplicate an existing conversion unit.
2) Edit the *Type of Conversion* field and change it to a new conversion type such as Power.
3) Change the *To Equation* and *From Equation* fields to just contain *m*.
4) Clear the Conversion Constant field.

For linear conversions, the *To Equation* field specifies how a given unit (*m*) is converted to the base units; the *From Equation* field specifies the inverse relationship; and *the Conversion Constant* field defines a constant used in the to/from conversion equations.

Let's look at an example from Figure 6. The base conversion unit for the Speed type is *Metres/Second*. The base type is easily recognized since both the *From Equation and To Equation* fields are set to *m*. If we convert from *Knots* to *Metres/Second*, we need to use the equation m*k so that 1 *Knot* equals k or 0.514444 *Metres/Second*. The inverse would require us to convert 1 *Metre/Second* to 1/k or 1.943846 *Knots*. There is nothing magical about the equations used to perform these conversions and they can even be nonlinear as the *Warp Factor* conversion unit illustrates. The asterisks ('*') indicate that the conversion will be exact. To guarantee that the final result is exact, both the originating and destination conversion units need to be exact. In other words, the conversion from *Inches/Second* to *Metres/Hour* will be exact because both units are exact but a conversion from *Inches/Second* to *Knots* will be inexact because the Knots units are not exact.



• Figure 6—Conversion Database Directory

Mortgage Calculation

Everyone has a mortgage. It is useful to be able to estimate your car payments and see how much interest you will pay before actually negotiating with a car dealer or bank to see what is affordable. An example mortgage for a $20,000 car which is repaid over four years at 4% per annum in monthly installments is shown in Figure 7. The only thing to remember when entering the loan parameters is that exactly three fields must be filled. The fourth will be automatically calculated when the mortgage table is displayed.

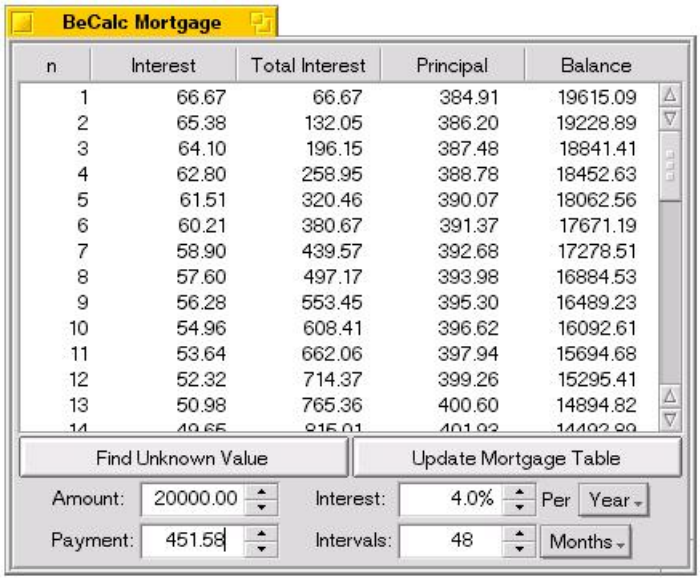
The mortgage display columns can be moved and resized just by dragging them with the mouse.

Printing a Mortgage List

1) Produce a desired mortgage list using the controls in the mortgage window.
2) Select the Print item in the File menu and select the Mortgage sub-menu item.

Exporting a Mortgage List

1) Produce a desired mortgage list using the controls in the mortgage window.
2) In the File menu, select the Export sub-menu and choose the Mortgage list item.
3) Select the desired export directory and enter the export filename in the file dialogue box. A tab-delimited version of the mortgage table will be created in a text file,



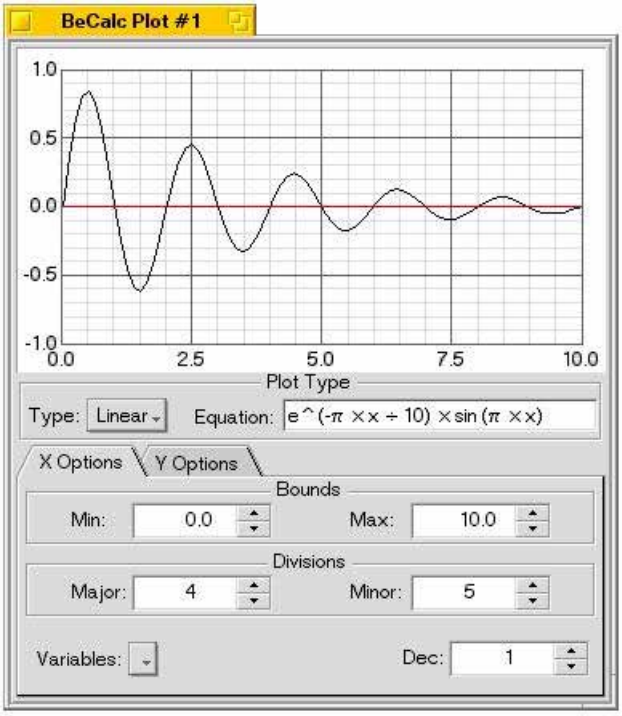
• Figure 7—Mortgage with $20000 amortized for 48 months at 4% per year

Plotting a Graph

This tutorial produces the two-dimensional plot shown in Figure 8. Set the *Degrees* pop-up menu to a *Radian* value. Enter the equation

$$e^{\frac{-\pi x}{10}} \cdot \sin \pi x$$

where the *x* is the required independent plot variable which BeCalc will vary over a given range to produce the output plot.



• Figure 8— Plot of the equation $e^{\frac{-\pi x}{10}} \cdot \sin \pi x$

1) Set the angular measure to radians using the angle pop-up menu.
2) Enter the equation using the following: e^, (, -, π, ×, x, ÷, 10,), ×, sin, (, π, ×, x,).
3) In the *Window* menu, select the *Open Plot* item to open the plot window and display this equation.
4) Change the maximum x-limit to 10 and the plot window will redisplay the full graph.

The plotted equation can be changed by editing the string displayed in the control just below the plot. The plot is updated when an <Enter> is typed on the keyboard.

The vertical axis (y-limits) can be changed either manually or automatically. Manual limits are entered into the y-min/max spin buttons and the plot is updated dynamically in response. The automatic limits scan the range of the current horizontal axis limits (x-limits) and update the y-min/max spin buttons with the corresponding minimum and maximum points.

Illegal x/y limits (e.g., minimum greater than a maximum) use valid default values for dynamic plot updates. This typically occurs during limit editing when no value is entered or during a plot of a partially entered limit.

Change the number of divisions and subdivisions plotted in either the x- or y-axes by clicking on the up or down arrow in the spin button. The plot is automatically updated when you finish setting the desired number of divisions or subdivisions.

Two more spin buttons change the number of decimal points in the x- and y-axis labels. As before, the plot automatically updates once a new selection is chosen.

BeCalc's main window menu shows all the available plot windows. Select the desired plot function to restore a closed plot window. All the limits and control settings are restored when the window is reopened. There is no limit to the number of plots which can be open at any time. However, only the first 25 plot windows will be automatically opened when *BeCalc* is started.

Exporting a Graph

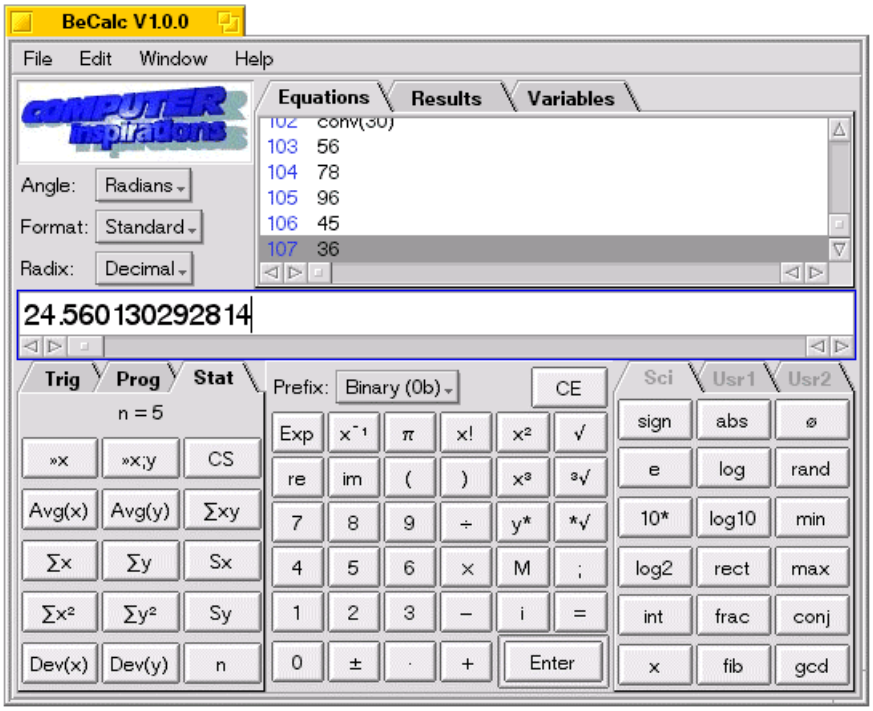
1) Create a plot and decorate it with the desired number of divisions and x/y intervals.
2) Click the left mouse in the plot area, and, holding it down drag the outline of the plot into an application which supports drag and drop of graphics and release the left mouse button to export the plot graphic. Alternatively, to create a file, drag the outline to a file window and drop it.

Statistics Population Characteristics

This example takes you through a session where we calculate the statistical functions related to a population with two independent variables. As a simple example, take the following five student grades:

56 78 96 45 36

1) Activate the statistics tab and clear the current state using the CS button.
2) Enter each of the above numbers in the entry control followed by a click on the »x button. The value of n will increase with each new sample.
3) Clear the display using CE and click on Avg(x) to display the average or mean value of the entered samples = 62.2.
4) Clear the display and click on the Dev(x) button to display the standard deviation in x = 21.967248….
5) Clear the display and click on the Sx button to display the variance = 482.56.
6) Click on any of the other keys to display other properties of the x population. Values of the y population will all be cleared.



• Figure 9—Statistical Functions

To enter two populations simultaneously, separate the numbers with a semicolon and click on the »x; y button.

Programmer Functions

A common use for a calculator is performing based-number conversions and calculations—often as a quick check to a coded algorithm or to generate input/output test vectors to verify a new function. *BeCalc* normally works with base ten numbers but can mix different numerical bases or can work entirely within one numerical base.

For example, try the following:

1) Select the hexadecimal mode from the radix popup menu.
2) Enter the following: 40F0 or 0o555 and 0b11101110010.
3) Click on the Enter key to calculate the result, 45F0.

In this example, the 40F0 is in hexadecimal notation (base 16), the 0o555 is in octal notation (base 8) and 0b11101110010 is in binary notation (base 2). Any number entered without a 0n prefix (n = b | t | o | d | u | x) will be assumed to be in the numerical base selected by the radix popup. Note as well that the *and* operator has higher precedence than the *or* operator. The Reference section on page 22 shows the operator priorities.

By selecting another base after a result has been calculated, the displayed number will be automatically converted into the new base. Variables are also displayed in the selected base although, internally, they are stored with full floating point precision. For example, if a variable contains the value 10.5, when the binary radix is selected, the number 1010 is displayed. The fraction is neither displayed nor used in any non-decimal radix calculations. Selecting a decimal radix again will display the fractional part of the number.

Next we use *BeCalc* to compute the Karnaugh map for a boolean expression Y = (A ⊕ B) + B·C + A·D + (B ⊕ D) shown in Table 3.

1) Set the calculator radix to binary.
2) Enter variables A, B, C, D representing four boolean vectors which we define as:

A = 0000_0000_1111_1111
B = 0000_1111_0000_1111
C = 0011_0011_0011_0011
D = 0101_0101_0101_0101

Note the use of the underscore as a grouping character. Other grouping characters are the apostrophe, comma, or decimal point. The leading zeros are not necessary and *BeCalc* will not display them. They are included only to clarify how these bit vectors were generated from Table 3.

3) Enter the equation: Y = (A xor B) or B and C or A and D or (B xor D). The '&' and '|' operators are equivalent to the 'and' and 'or' operators, respectively.
4) The result should be 0000111100001111. You can also view the result with grouped digits by turning on the underscore character as a grouping separator in the preferences.
5) Now, by dragging the equation back to the entry control and modifying it, you can generate the Karnaugh map of any four-variable function by reusing the test vectors (i.e., variables A, B, C, and D) with the modified equation.

When entering the boolean expression for Y it was necessary to provide brackets around the exclusive or operators because they have the same precedence as the inclusive or operator. However, no brackets were needed around the and operator since it has higher

precedence than the or operator. Experiment by removing the brackets and inserting some around the and operators to see the effects.

A	B	C	D	Y
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**0**
0	0	1	1	**1**
0	1	0	0	**1**
0	1	0	1	**1**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**1**
1	0	1	0	**1**
1	0	1	1	**1**
1	1	0	0	**1**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**1**

• Table 3—Karnaugh map of $Y = (A \oplus B) + B{\cdot}C + A{\cdot}D + (B \oplus D)$

Radix Number Accuracy

Based number accuracy is dependent on the number of digits selected in the customization dialogue (see below). For example, assume that numbers are accurate to 16 digits. In this case, a binary number must be exactly representable without requiring an exponent. Thus, the maximum number would be $1\times10^{16} - 1$ which gives approximately $\log_2(1\times10^{16}) = 53$ bits. This number is rounded down to the nearest multiple of four bits or 52 bits. For reference purposes, a 200-digit accuracy corresponds to about 664 bits!

Determine the current maximum number of available bits for a given numerical setting with this expression: cnt(-1). The reason this works is that *BeCalc* actually represents based negative numbers in two's complement format—the same way your computer also works—so that a negative number is represented by a unsigned integer. The integer –1 is represented by the largest possible integer, thus, by counting its bits, you determine the maximum number of bits for a given numerical resolution. You can see the equivalent bit representation of any negative number by converting that number to either hexadecimal or binary notation using the radix pop-up menu. Also try changing the number of digits to see the effect on the bit representation.

Customizations

Next we customize *BeCalc* to suit your specific preferences for different numeric accuracies and fixed point notation, list preferences, and much more. This tutorial familiarizes you with the preferences window () and some of the available options.

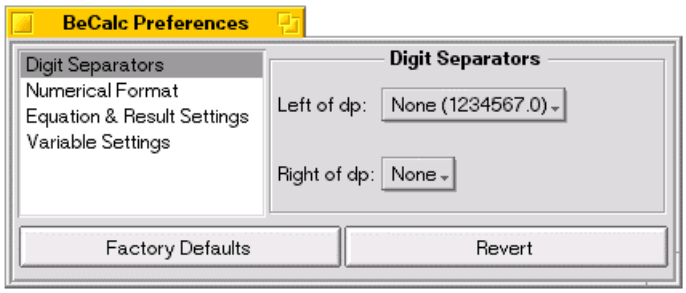
Digit Separators

Since *BeCalc* is capable of working with very large numbers, a method is needed of splitting up the digits into smaller groupings to provide reference points when interpreting results or entering a large number. The digit separators consisting of the underscore, apostrophe, comma, and decimal point provide such a mechanism.

With the exception of the decimal point, any other grouping separator can be freely used when entering any number. However, the decimal point is only valid when the default digit separator has also been chosen to be a decimal point. In this case, the comma serves as a units/fractions separation point in place of the decimal point. Separators are inserted every three digits for decimal numbers and every four digits for numbers in other bases.

To select a digit separator:

1) Open the preferences window by selecting the Settings option in the Edit menu.
2) Click on the *Digit Separators* list item in the preferences window.
3) Select the desired separator for digits to the left of the decimal point with the first pop-up menu and the desired fractional digit separator to be either the same as the first selection or to have no fractional digit separator.



• Figure 10—Digit Separators

Numerical Format

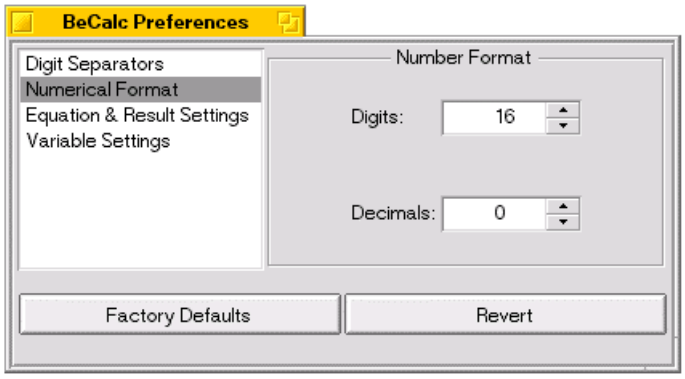
BeCalc's powerful numerical algorithms provide several formatting options:

- choice of scientific, engineering, or standard notation
- based numbers with radices from 2 to 16.
- floating point or fixed point numbers
- selection of the displayed number accuracy

The first two options are available in *BeCalc*'s main window via the format and radix pop-up menus.

Floating point numbers are selected by using the decimals spin button in the numerical format preferences window to set a value of zero. Any non-zero value will produce that number of decimal places in a displayed number. If the number of decimal places exceed the available number of digits only the maximum possible number of decimal places will be displayed.

The accuracy of *BeCalc*'s calculations can be altered with the digits spin button. By selecting more digits, additional accuracy will be used during calculations and more digits will be displayed as a result. Increasing the number of digits will require more computational resources so a trade-off based on your computer's performance may be necessary.

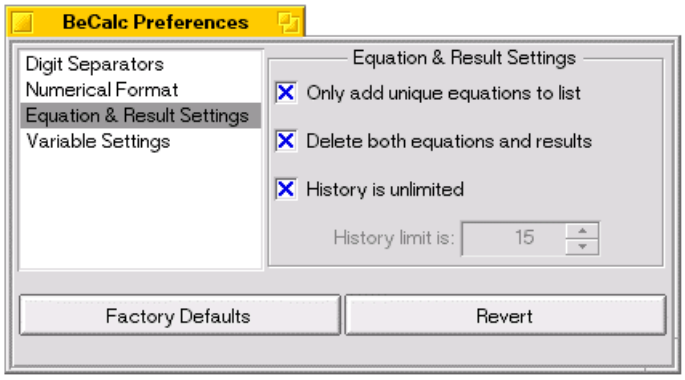


• Figure 11—Numerical Format

Equation and Result Settings

Several options are available:

- Add only unique equations to the equation history list by checking the first box. If a new result is computed, the result window's contents corresponding to equation will be updated.
- Select the second box to always delete both equations and the associated result in the equation and result history lists. By default, this option is disabled so a requester will always appear to give a choice of whether the result associated with a selected equation should also be deleted.
- Selecting the History is unlimited box allows an unlimited number of items in the equation and result history lists. When unselected, the corresponding roll button can be used to set an exact limit. If the specified number of items is exceeded, older results will be deleted to make room for new items. The default setting is to provide an unlimited number of list entries.

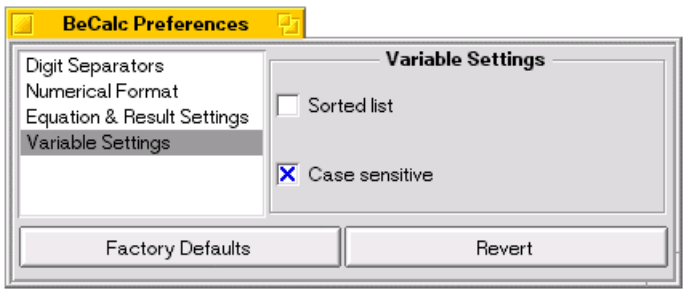


• Figure 12—Equation and Result Settings

Variable Settings

The displayed variable list can be automatically sorted in alphabetical order by checking the sorted list box. The default is an unsorted list.

The case sensitivity of variables can be altered by selecting the case sensitive box. When checked, variables are case sensitive. The default is to use case sensitive variables. Note that switching from case sensitive to case insensitive may prevent *BeCalc* from distinguishing between two or more variables with the same spelling with different upper- and lower-case letters. The first defined variable will be chosen for updates in the case of such a conflict.



• Figure 13—Variable Settings

Chapter

4

BeCalc User Manual

Installation Instructions And Guided Tutorials

Reference

This chapter provides a convenient reference of all *BeCalc* operations and functions. Operations are listed in order of priority where the highest-priority operators appear first. Functions are listed in alphabetical order. In some cases multiple operators and functions exist for the same operations. This redundancy is for the user's convenience since some operator symbols are not easily accessible from standard keyboards. For each operator or function, a brief description and an illustrative example are shown.

()	Parentheses. Grouped expressions are calculated first. (3+4)*(5+6) = 77.
^, **	Power operators. 10^5 = 100000.
²	Squared operator. $10^2 = 100$.
³	Cubed operator. $10^3 = 1000$.
-1	Reciprocal operator. $9^{-1} = 0.11111111\ldots$
√	Square root operator. $\sqrt{9} = 3$.
³√	Cube root operator. $\sqrt[3]{27} = 3$.
***√**	Nth root operator. $8 *\sqrt{} 256 = 2$.
***, ×**	Multiplication operators. 123 × 456 = 56088.
/, ÷	Floating point division operators. 1 ÷ 9 = 0.11111111…
&	Logical and operator. 00FF & 5555 = 55.
+	Addition operator. 9999 + 1234 = 11233.
-, –	Subtraction operator. 9999 - 1234 = 8765.
|	Logical or operator. 00FF | 5555 = 55FF.
~	Logical not operator. ~FF5A = FFFF…00A5.
ø	Return the polar angle for an imaginary number. ø(1 + i) = 45.
<var>	Return the contents of a variable. A variable begins with an alphabetic character followed by zero or more alphanumeric characters. M0 = 0.
<var> =	Define or update a variable definition. Any legal equation can be used to define a variable. The equation is evaluated and stored into the variable location. M0 = (23.45 + sin 45) × 4.
π	Constant pi accurate to over 200 decimal places. π = 3.141592654…
abs	Absolute value or magnitude function. abs(-100) = 100, abs(1+i) = 1.4142…
and	Logical and operator. 00FF and 5555 = 55.
acos	Trigonometric inverse cosine. acos(0) = 90.
acosh	Hyperbolic inverse cosine. acosh(1) = 0.
asin	Trigonometric inverse sine. asin(1) = 90.

asinh Hyberbolic inverse sine. asinh(1) = 0.881373587…

asr Shift a number right by the specified bit count. Shifted bits are set to the number's sign bit. -1 asr 3 = -1.

acot Trigonometric inverse cotangent. acot(1) = 45.

acoth Hyperbolic inverse cotangent. acoth(1) = 0.5.

atan Trigonometric inverse tangent. atan(1) = 45.

atanh Hyberbolic inverse tangent. atanh(0.5) = 0.549306144…

cbit Clear the specified bit of a number to '0'. 15 cbit 0 = 14.

cbrt Cube root operator. cbrt(27) = 3.

conj Returns the complex conjugate of an imaginary number. conj(1 + i) = 1 - i.

conv Converts from the *from* units to the *to* units. If the *from* units are in centimeters and the *to* units are in inches: conv(1) = 0.3937.

convi Converts from the *to* units to the *from* units. If the *from* units are in centimeters and the *to* units are in inches: convi(1) = 2.5400.

count Counts the number of bits set to logical 1 in a number. count(100) = 3.

cos Trigonometric cosine function. cos(90) = 0.

cos^{-1} Trigonometric inverse cosine. $\cos^{-1}(0) = 90$.

cosh Hyberbolic cosine function. cosh(0) = 1.

cosh^{-1} Hyberbolic inverse cosine. $\cosh^{-1}(1) = 0$.

cot Trigonometric cotangent. cot(45) = 1.

coth Hyperbolic cotangent. coth(2.1639534) = 0.5.

cot^{-1} Trigonometric inverse cotangent. $\cot^{-1}(1) = 45$.

coth^{-1} Hyperbolic inverse cotangent. $\coth^{-1}(0.5) = 2.1639534$.

div Integral division. 123 div 56 = 2.

e Base of the natural logarithm. e = 2.718281828…

fib Display the nth Fibonacci number where $n \geq 0$. fib(20) = 6765.

frac Fractional part of a number. frac(1.5) = 0.5.

gcd Greatest common denominator. gcd(1000, 1200) = 200.

int Integral part of a number. int(1.5) = 1.

i Imaginary number defined as $\sqrt{-1}$. (56 + 23i)·(5 -16i) = 648 - 781i.

im Returns the imaginary part of a number. im(1+25i) = 25.

log Natural logarithm or base e exponent which gives the input. log(e) = 1.

log10 Common logarithm or base 10 exponent which gives the input. log10(100) = 2

log2 Base 2 exponent which gives the input. log2(1073741824) = 30.

max Larger of two numbers. max(-56; 23) = 23.

min Smaller of two numbers. min(-56; 23) = -56.

not Logical not operator. mot FF5A = FFFF…00A5.

mod Modulo operator. 56 mod 5 = 1.

nand Logical nand operator. 56 nand 5 = 0FFFF...FFFA.

nor Logical nor operator. 56 nor 5 = 0FFFF...FFC2.

nCr Calculates n! / (n-r)!. 12 nCr 10 = 66.

nPr Calculates n! / r! (n-r)!. 12 nPr 10 = 239500800.

or Logical or operator. 00FF or 5555 = 55FF.

pi Constant pi accurate to over 200 decimal places. pi = 3.141592654…

r Returns the magnitude of an imaginary number. r(1 + i) = 1.41421…

rand Random number function. rand = 0.99118760542628735402…

re Returns the real part of a number. re(1+25i) = 1.

rect Converts an imaginary number in polar coordinates to rectangular coordinates. To convert $1 \angle 45$ we have: 1 rect 45 = 1 + i.

root Nth root operator. 8 root 256 = 2.

	Constant pi accurate to over 200 decimal places. π = 3.141592654…
rol	Rotate a number left by the specified number of bits. The highest bits are shifted into the lowest bit positions. 8 rol 3 = 64.
ror	Rotate a number right by the specified number of bits. The lowest bits are shifted into the highest bit positions. 8 ror 3 = 1.
sbit	Set the specified bit of a number to '1'. 8 sbit 0 = 9.
shl	Shift a number left by the specified bit count. Shifted bits are 0. 1 shl 3 = 8.
shr	Shift a number right by the specified bit count. Shifted bits are 0. 8 shr 3 = 1.
sign	Sign of a number. Negative is –1, zero is 0, positive is 1. sign (-56) = -1.
sin	Trigonometric sine function. sin(90) = 1.
$\sin^{-1}$	Trigonometric inverse sine. $\sin^{-1}(1) = 90$.
sinh	Hyberbolic sine function. sinh(0.521095305) = 0.5.
$\sinh^{-1}$	Hyberbolic inverse sine. $\sinh^{-1}(0.5) = 0.521095305…$
sqrt	Square root function. sqrt (9) = 3.
tan	Trigonometric tangent function. tan(45) = 1.
$\tan^{-1}$	Trigonometric inverse tangent. $\tan^{-1}(1) = 45$.
tanh	Hyberbolic tangent function. tanh(0) = 1.
$\tanh^{-1}$	Hyberbolic inverse tangent. $\tanh^{-1}(0.5) = 0.549306144…$
tbit	Toggle the specified bit of a number. 15 tbit 0 = 14.
xor	Exclusive Or operator. 5555AAAA xor 0AAAAFFFF = 0FFFF5555
»x	Add to statistical x population. Enter a number or expression and click on this key to add a statistical datum to the x population. The n display will increment.
»x;y	Add to statistical x population. Enter a number or expression and click on this key to add a statistical datum to the x population. The n display will increment.
CS	Clear statistic state. n will be reset to 0 and all internal state variables will be cleared.
Avg(x)	Inserts the average of the x population at the current cursor position.
Avg(y)	Inserts the average of the y population at the current cursor position.
Σxy	Inserts the sum of all x·y values at the current cursor position.
Σx	Inserts the sum of all x values at the current cursor position.
Σy	Inserts the sum of all y values at the current cursor position.
Σx	Inserts the sigma value of x population at the current cursor position.
Sy	Inserts the sigma value of y population at the current cursor position.
Σx^2	Inserts the sum of all x^2 values at the current cursor position.
Σy^2	Inserts the sum of all y^2 values at the current cursor position.
Dev(x)	Inserts the standard deviation in x at the current cursor position.
Dev(y)	Inserts the standard deviation in y at the current cursor position.
n	Inserts the total number of x/y values at the current cursor position.